Boosting with Multiple Sources

Anonymous Author(s)
Affiliation
Address
email

Abstract

We study the problem of learning accurate ensemble predictors, in particular boosting, in the presence of multiple source domains. We show that the standard convex combination ensembles in general cannot succeed in this scenario and adopt instead a domain-weighted combination. We introduce and analyze a new boosting algorithm, MULTIBOOST, for this scenario and show that it benefits from favorable theoretical guarantees. We also report the results of several experiments with our algorithm demonstrating that it outperforms natural baselines on multi-source text-based, image-based and tabular data. Additionally, we present an extension of our algorithm to the federated learning scenario and report favorable experimental results for that setting as well.

1 Motivation

Ensemble methods such as Bagging, AdaBoost, Stacking, error-correction techniques, Bayesian averaging, AdaNet or other adaptive methods for learning neural networks are general machine learning techniques used to combine several predictors to devise a more accurate one (Breiman, 1996; Freund and Schapire, 1997; Smyth and Wolpert, 1999; MacKay, 1991; Freund et al., 2004; Cortes et al., 2017). These techniques are often very effective in practice and benefit from favorable margin-based learning guarantees (Schapire et al., 1997). These algorithms assume access to a training sample drawn from the target distribution. But, in many applications, the learner receives labeled data from multiple source domains that it seeks to use to find a good predictor for target domains, which are typically assumed to be mixtures of source distributions (Hoffman et al., 2018; Mansour et al., 2008; Muandet et al., 2013; Xu et al., 2014; Hoffman et al., 2012; Saito et al., 2019; Wang et al., 2019a). How can we generalize ensemble methods such as boosting to this scenario?

Several related problems have been tackled in previous work. In the special case of a single target domain, boosting solutions have been derived (Dai et al., 2007; Yao and Doretto, 2010). These algorithms have been further improved by boosting base predictors jointly with source features (or feature *views*) that are predictive in the target (Yuan et al., 2017; Cheng et al., 2013; Zhang et al., 2014; Xu and Sun, 2012, 2011). Such methods have been widely adopted in various domain adaptation scenarios for different types of data. For example, Huang et al. (2010, 2012) showed that by selecting a base learner jointly with a feature that is predictive across multiple domains at every boosting step, one can achieve higher accuracy than standard transfer learning methods. Moreover, the margin provided by boosting-style algorithms can aid in transfer learning where target domain is unlabelled. Habrard et al. (2013) have developed an algorithm that jointly minimizes the source domain error and margin violation proportion on the target domain. Wang et al. (2019a) have demonstrated that boosting classifiers from different domains can be carried out online. Taherkhani et al. (2020) and Becker et al. (2013) have shown that multi-source boosting can be combined with Deep Neural Networks for multi-task learning on large scale datasets.

We give a more detailed discussion of this related prior work in Appendix A. However, as we demonstrate in this paper, several critical questions related to the presence of multiple domains have

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

not been fully addressed by existing boosting methods from both the algorithmic and the theoretical point of view. First, what should be the form of the ensemble solutions? We show that, in general, the standard convex combinations used in much of previous work may not succeed in this problem (Section 2). Instead, we put forward Q-ensembles, which are convex combinations weighted by a domain classifier Q, that is, $\mathsf{Q}(k|x)$ is the conditional probability of domain k given input point x. This crucially differentiates our work from that of past studies. Our learning scenario strictly generalizes previous algorithms. In the special case of a single domain, the form of our ensemble solutions coincides with that of the familiar ensembles used in AdaBoost.

Second, what should be the form of the objective? Unlike existing boosting methods which optimize for a specific target distribution or domain, we seek a solution that is accurate for any mixture of the source distributions, where the mixture weights may be constrained to be in a subset of the simplex. This conforms with the learning goal in this scenario where the target distribution is typically assumed to be a mixture of the source ones and is further inspired by the formulation of the *agnostic loss* in the related context of federated learning (Mohri, Sivek, and Suresh, 2019). Additionally, this formulation can be viewed as more risk-averse and robust, and it also ensures more favorable fairness properties, since it does not favor any distribution possibly biased towards some subset of the source domains. In Section 3, we introduce and analyze an algorithm, MULTIBOOST, that is based on such an objective. This further distinguishes our boosting algorithm from related prior work.

Third, in Section 4, we present a theoretical analysis of our MULTIBOOST algorithm, including margin-based generalization bounds that hold for any target mixture of the source distributions. In Appendix F, we derive finer and more flexible learning bounds that can provide more favorable guarantees, in particular when the family of target mixtures is more constrained.

Fourth, in Section 5, we describe FEDMULTIBOOST, an extension of our MULTIBOOST algorithm adapted to the distributed learning scenario of *federated learning*, which is increasingly important in a wide array of applications (Kairouz et al., 2021), including healthcare (Brisimi et al., 2018). FEDMULTIBOOST allows the server to train ensemble models that benefit from user data, while the data remains with that users. We provide a detailed description of our algorithm, as well as a brief analysis of the communication costs, which are critical in this scenario. Appendix G contains experimental results comparing FEDMULTIBOOST with several baselines. Finally, in Section 6, we report the results of extensive experiments with our MULTIBOOST algorithm. We start with a detailed description of the learning scenario we consider.

2 Learning scenario

Let $\mathcal{X}$ denote the input space and $\mathcal{Y} = \{-1, +1\}$ be the output space, associated to binary classification. We assume the learner receives labeled samples from p source domains, each defined by a distribution $\mathcal{D}_k$ over $\mathcal{X} \times \mathcal{Y}$, $k \in [p]$. We denote by $S_k = ((x_{k,1}, y_{k,1}), \ldots, (x_{k,m_k}, y_{k,m_k})) \in \left(\mathcal{X} \times \mathcal{Y}\right)^{m_k}$ the labeled sample of size m_k received from source k, which is drawn i.i.d. from $\mathcal{D}_k$. For any function $f \colon \mathcal{X} \to \mathbb{R}$ and distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, let $\mathcal{L}(\mathcal{D}, f)$ be the expected loss of f, that is $\mathcal{L}(\mathcal{D}, f) = \mathbb{E}_{(x,y)\sim\mathcal{D}}[\ell(f(x), y)]$, where ℓ is the binary loss $\ell(f(x), y) = \mathbb{I}(yf(x) \leq 0)$.

For any $k \in [p]$, let $\mathcal{H}_k$ denote a hypotheses set of functions mapping from $\mathcal{X}$ to $[-1, +1]$, $|\mathcal{H}_k| = N_k$. The objective of the learner is to find an accurate predictor f for any target distribution $\mathcal{D}_\lambda$ that is a mixture of the source distributions, where λ may be in a subset Λ of the simplex. Specifically, $\mathcal{D}_\lambda = \sum_{k=1}^{p} \lambda_k \mathcal{D}_k$, with $\lambda = (\lambda_1, \ldots, \lambda_p)$, the set of distributions over the p domains. This leads to the following *agnostic loss* of a predictor f (Mohri et al., 2019):

$$\mathcal{L}(\mathcal{D}_\Lambda, f) = \max_{\lambda \in \Lambda} \mathcal{L}(\mathcal{D}_\lambda, f). \tag{1}$$

To create a predictor f, the learner seeks an ensemble of functions from the base classes $\mathcal{H}_k$. What should be the form of such ensembles? A natural solution is a convex combinations of the form

$$f = \sum_{l=1}^{p} \sum_{j=1}^{N_l} \alpha_{l,j} h_{l,j}, \tag{2}$$

where $h_{l,j} \in \mathcal{H}_l$ and $\alpha_{l,j} \geq 0$, $\sum_{l=1}^{p} \sum_{j=1}^{N_l} \alpha_{l,j} = 1$, as in prominent algorithms such as BAGGING (Breiman, 1999) or ADABOOST (Freund and Schapire, 1997). It is not hard to show, however, that for some distributions, any such convex combination of base predictors would lead to a poor solution.

Proposition 1. *There exist distributions $\mathcal{D}_1$ and $\mathcal{D}_2$ and hypotheses h_1 and h_2 with $\mathcal{L}(\mathcal{D}_1, h_1) = 0$ and $\mathcal{L}(\mathcal{D}_2, h_2) = 0$ such that $\mathcal{L}\big(\frac{1}{2}(\mathcal{D}_1 + \mathcal{D}_2), \alpha h_1 + (1-\alpha)h_2\big) \geq \frac{1}{2}$ for any $\alpha \in [0, 1]$,*

see Appendix B. The base predictors h_1, h_2 are each perfect for their respective domains, but unlike the common case of a single source domain, standard convex combinations in general may perform poorly. Thus, we will consider instead the following form for the ensembles of base predictors:

$$\forall x \in \mathcal{X}, \quad f(x) = \sum_{l=1}^{p} \sum_{j=1}^{N_l} \alpha_{l,j} \mathsf{Q}(l|x) h_{l,j}(x), \tag{3}$$

where $\mathsf{Q}(l|x)$ denotes the conditional probability of domain l given x. This helps us account for that the learner combines base predictors from different domains. For a given point x, experts from the domains where x is more likely to appear are allocated more weight in the voted combination. The following result further substantiates the hope for this method to succeed.

Proposition 2. *For the same distributions $\mathcal{D}_1$ and $\mathcal{D}_2$ and hypotheses h_1 and h_2 as in Proposition 1, the equality $\mathcal{L}(\mathcal{D}_\lambda, (\alpha \mathsf{Q}(1|\cdot)h_1 + (1-\alpha)\mathsf{Q}(2|\cdot)h_2)) = 0$ holds for any $\lambda \in \Delta$ and any $\alpha \in (0, 1)$,*

see proof in Appendix B. Thus, our Q-ensembles can be substantially more effective in the multiple-source adaptation problem considered. Note, that in the special case of a single domain, the Q-combinations coincide with standard convex combinations, since $\mathsf{Q}(k|x) = 1$ for all x.

As in the standard boosting scenario, we will adopt a weak-learning assumption. However, our assumption here must hold for each source domain: for each domain $k \in [p]$ and any distribution D over S_k, there exists a base classifier $h \in \mathcal{H}_k$ such that the weighted loss of h is γ-better than random: $\frac{1}{2}[1 - \mathbb{E}_{i\sim\mathsf{D}}[y_{k,i}h(x_{k,i})] \leq \frac{1}{2} - \gamma$, for some edge value $\gamma > 0$. This is equivalent to the existence of a good *rule of thumb* for each domain. Unlike the standard scenario, however, here we seek a Q-ensemble and further require it to be accurate for any target mixture $\mathcal{D}_\lambda$. In the next section, we present an algorithm, MULTIBOOST, for deriving an accurate Q-ensemble for any target mixture domains that belongs to the convex combination of the source domains.

3 Algorithm

Let Φ be a convex, increasing and differentiable function such that $u \mapsto \Phi(-u)$ upper-bounds the binary loss $u \mapsto 1_{u\leq 0}$. Φ could be the exponential function as in ADABOOST or the logistic function, as in logistic regression. Using Φ to upper-bound the agnostic loss leads to the following objective function for an ensemble f defined by (3) for any $\alpha = (\alpha_{l,j})_{(l,j)\in[p]\times[N_l]} \geq 0$:

$$F(\alpha) = \max_{\lambda\in\Lambda} \sum_{k=1}^{p} \frac{\lambda_k}{m_k} \sum_{i=1}^{m_k} \Phi\bigg(-y_{k,i} \sum_{l=1}^{p} \sum_{j=1}^{N_l} \alpha_{l,j} \mathsf{Q}(l|x_{k,i}) h_{l,j}(x_{k,i})\bigg). \tag{4}$$

Since a convex function composed with an affine function is convex and a sum of convex functions is convex, F is convex as the maximum of a set of convex functions. In this section, we will consider the case where the set Λ coincides with the full simplex, that is $\Lambda = \Delta$. Our algorithm applies coordinate descent to this objective, as with ADABOOST and other boosting-type algorithms. While convex, F is not differentiable and, in general, coordinate descent may not succeed in such cases (Tseng, 2001; Luo and Tseng, 1992). This is because the algorithm may be *stuck* at a point where no progress is possible along any of the axis directions, while there exists a favorable descent along some other direction. However, we will show that, under the weak-learning assumption we adopted, there always exists a main direction, along which a descent progress is possible. Alternatively, one can replace the maximum with a *soft-max*, that is the $(x_1, \ldots, x_k) \mapsto \frac{1}{\mu}\log(\sum_{k=1}^{p} e^{\mu x_k})$ for $\mu > 0$, which leads to a continuously differentiable function with a close approximation for η sufficiently large.

Description. Let α_{t-1} denote the value of the parameter vector $\alpha = (\alpha_{l,j})$ at the end of the $(t-1)$th iteration and f_{t-1} the corresponding function: $f_{t-1} = \sum_{l=1}^{p}\sum_{j=1}^{N_l} \alpha_{t-1,l,j}\mathsf{Q}(l|\cdot)\, h_{l,j}$. Coordinate descent at iteration t consists of choosing a direction $\mathbf{e}_{q,r}$ corresponding to base classifier $h_{q,r}$ and a step value $\eta > 0$ to minimize $F(\alpha_{t-1} + \eta\, \mathbf{e}_{q,r})$. To select a direction, we consider the subdifferential of F along any $\mathbf{e}_{q,r}$. Observe that, since $\Lambda = \Delta$, for any α, we can write $F(\alpha) = \sum_{k=1}^{p} F_k(\alpha)$, with $F_k(\alpha) = \sum_{k=1}^{p} \frac{1}{m_k} \sum_{i=1}^{m_k} \Phi\big(-y_{k,i} \sum_{l=1}^{p}\sum_{j=1}^{N_l} \alpha_{l,j}\mathsf{Q}(l|x_{k,i})h_{l,j}(x_{k,i})\big)$. Since functions F_k

are differentiable, by the subdifferential calculus for the maximum of functions, the subdifferential of F at α_{t-1} along the direction $\mathbf{e}_{q,r}$ is given by:

$$\partial F(\alpha_{t-1}, \mathbf{e}_{q,r}) = \operatorname{conv}\{F'_k(\alpha_{t-1}, \mathbf{e}_{q,r})\colon k \in \mathcal{K}_t\},$$

where $F'_k(\alpha_{t-1}, \mathbf{e}_{q,r})$ is the directional derivative of F_k at α_{t-1} along the direction $\mathbf{e}_{q,r}$ and where $\mathcal{K}_t = \{k \in [p]\colon F_k(\alpha_{t-1}) = F(\alpha_{t-1})\}$. We will therefore consider the direction $\mathbf{e}_{q,r}$ with the largest absolute directional derivative value $|F'_k(\alpha_{t-1}, \mathbf{e}_{q,r})|$, $k \in \mathcal{K}_t$, but will restrict ourselves to $q = k$ since, as we shall see, that will suffice us to guarantee a non-zero directional gradient. To do so, we will express $F'_k(\alpha_{t-1}, \mathbf{e}_{k,r})$ in terms of the distribution $\mathsf{D}_t(k, \cdot)$ over S_k defined by $\mathsf{D}_t(k, i) = \frac{\Phi'(-y_{k,i} f_{t-1}(x_{k,i}))}{Z_{t,k}}$, with $Z_{t,k} = \sum_{i=1}^{m_k} \Phi'(-y_{k,i} f_{t-1}(x_{k,i}))$, for all $i \in [m_k]$:

$$F'_k(\alpha_{t-1}, h_{k,r}) = \frac{1}{m_k} \sum_{i=1}^{m_k} -y_{k,i} \mathsf{Q}(k|x_{k,i}) h_{k,r}(x_{k,i}) \Phi'\big(-y_{k,i} f_{t-1}(x_{k,i})\big) = \frac{Z_{t,k}}{m_k}[2\epsilon_{t,k,r} - 1],$$

where $\epsilon_{t,k,r} = \frac{1}{2}\big[1 - \mathbb{E}_{i\sim \mathsf{D}_t(k,\cdot)}[y_{k,i} \mathsf{Q}(k|x_{k,i}) h_{k,r}(x_{k,i})]\big]$ denotes the *weighted error* of $\mathsf{Q}(k|\cdot) h_{k,r}$. For any $s \in [m_k]$, since $x_{k,s}$ is a sample drawn from $\mathcal{D}_k$, we have $\mathsf{Q}(k|x_{k,s}) > 0$ and therefore we have: $\overline{\mathsf{Q}}_{t,k} = \sum_{s=1}^{m_k} \mathsf{D}_t(k, s) \mathsf{Q}(k|x_{k,s}) > 0$. Thus, we can write $\mathbb{E}_{i\sim \mathsf{D}_t(k,\cdot)}[y_{k,i} \mathsf{Q}(k|x_{k,i}) h_{k,r}(x_{k,i})]$ as

$$\sum_{i=1}^{m_k} \frac{\mathsf{D}_t(k, i) \mathsf{Q}(k|x_{k,i}) y_{k,i} h_{k,r}(x_{k,i})}{\overline{\mathsf{Q}}_{t,k}} \overline{\mathsf{Q}}_{t,k}, = \mathop{\mathbb{E}}_{i\sim \mathsf{D}'_t(k,\cdot)}[y_{k,i} h_{k,r}(x_{k,i})]\, \overline{\mathsf{Q}}_{t,k},$$

where $\mathsf{D}'_t(k, i) = \frac{\mathsf{D}_t(k,i) \mathsf{Q}(k|x_{k,i})}{\overline{\mathsf{Q}}_{t,k}}$. By our weak-learning assumption, there exists $r \in N_k$ such that $\mathbb{E}_{i\sim \mathsf{D}'_t(k,\cdot)}[y_{k,i} h_{k,r}(x_{k,i})] \geq \gamma > 0$. For that choice of r, we have $\epsilon_{t,k,r} < \frac{1}{2} - \overline{\gamma}$, with $\overline{\gamma} = \gamma \overline{\mathsf{Q}}_{t,k} > 0$. In view of that, it suffices for us to search along the directions $h_{k,r}$ and we do not need to consider the directional derivative of F_k along directions $h_{q,r}$ with $q \neq k$. The direction chosen by our coordinate descent algorithm is thus defined by: $\operatorname{argmax}_{k\in\mathcal{K}_t, r\in[N_k]} \frac{Z_{t,k}}{m_k}[1 - 2\epsilon_{t,k,r}]$. Given the direction $\mathbf{e}_{k,r}$, the optimal step value η is $\operatorname{argmin}_{\eta>0} F(\alpha_{t-1} + \eta \mathbf{e}_{k,r})$. The pseudocode of our algorithm, MULTIBOOST, is provided in Figure 1. In the most general case, η can be found via a line search or other numerical methods.

Step size. In some special cases, the line search can be executed using a simpler expression by using an upper bound on $F(\alpha_{t-1} + \eta \mathbf{e}_{k,r})$. Using the convexity of Φ, since $y_{l,i} \mathsf{Q}(k|x_{l,i}) h_{k,r}(x_{l,i}) = \frac{1 + y_{l,i} \mathsf{Q}(k|x_{l,i}) h_{k,r}(x_{l,i})}{2} \cdot (+1) + \frac{1 - y_{l,i} \mathsf{Q}(k|x_{l,i}) h_{k,r}(x_{l,i})}{2} \cdot (-1)$, the following holds for all $\eta \in \mathbb{R}$:

$$\begin{aligned}\Phi(-y_{l,i} f_{t-1}(x_{l,i}) - \eta\, y_{l,i} \mathsf{Q}(k|x_{l,i}) h_{k,r}(x_{l,i})) \leq{}& \tfrac{1 + y_{l,i} \mathsf{Q}(k|x_{l,i}) h_{k,r}(x_{l,i})}{2} \Phi(-y_{l,i} f_{t-1}(x_{l,i}) - \eta) \\ &+ \tfrac{1 - y_{l,i} \mathsf{Q}(k|x_{l,i}) h_{k,r}(x_{l,i})}{2} \Phi(-y_{l,i} f_{t-1}(x_{l,i}) + \eta).\end{aligned}$$

In the case of exponential and logistic functions, the following upper bounds can then be derived.

Lemma 1. *For any $k \in [p]$, the following upper bound holds when Φ is the exponential or the logistic function:*

$$F(\alpha_{t-1} + \eta \mathbf{e}_{k,r}) \leq \max_{l\in[p]} \frac{Z_{t,l}}{m_l} \big[(1 - \epsilon_{t,l,k,r}) e^{-\eta} + \epsilon_{t,l,k,r} e^{\eta}\big],$$

where $\epsilon_{t,l,k,r} = \frac{1}{2}\big[1 - \mathbb{E}_{i\sim \mathsf{D}_t(l,\cdot)}[y_{l,i} \mathsf{Q}(k|x_{l,i}) h_{k,r}(x_{l,i})]\big]$.

For any k, function $\eta \mapsto (1 - \epsilon_{t,l,k,r}) e^{-\eta} + \epsilon_{t,l,k,r} e^{\eta}$ reaches its minimum for $\eta = \frac{1}{2} \log \frac{1 - \epsilon_{t,l,k,r}}{\epsilon_{t,l,k,r}}$. When the maximum is achieved for $l = k$, the solution coincides with the familiar expression of the step size $\eta_t = \frac{1}{2} \log \frac{1 - \epsilon_{t,k,r}}{\epsilon_{t,k,r}}$ used in ADABOOST.

Q-function. The conditional probability functions $\mathsf{Q}(k|\cdot)$ are crucial to the definition of our algorithm. As pointed out earlier, Q-ensembles can help achieve accurate solutions in some cases that cannot be realized using convex combinations. Furthermore, for any $k \in [p]$, since $\mathsf{D}_t(k, \cdot)$ is a distribution over the sample S_k, it is natural to assume that for any $j \neq k$ we have $\mathbb{E}_{s\sim \mathsf{D}_t(k,\cdot)}[\mathsf{Q}(k|x_{k,s})] \geq \mathbb{E}_{s\sim \mathsf{D}_t(k,\cdot)}[\mathsf{Q}(j|x_{k,s})]$. This implies the following lower bound: $\mathbb{E}_{s\sim \mathsf{D}_t(k,\cdot)}[\mathsf{Q}(k|x_{k,s})] \geq \frac{1}{p}$, which in turn implies $\overline{\gamma} \geq \frac{\gamma}{p}$, since for any $x \in \mathcal{X}$, $\sum_{j=1}^{p} \mathsf{Q}(j|x) = 1$. In the special case where all domains coincide, we have $\mathsf{Q}(k|x_{k,s}) = \frac{1}{p}$ for all s and this lower bound is reached. At another extreme, when all domains admit distinct supports, we have $\mathsf{Q}(k|x_{k,s}) = 1$ for all $s \in [m_k]$ and thus $\overline{\gamma} = \gamma$.

In practice, we do not have access to the true conditional probabilities $\mathsf{Q}(k|\cdot)$. Instead, we can derive accurate estimates $\widehat{\mathsf{Q}}(k|\cdot)$ using large unlabeled samples from the source domains, the *label* used for training being simply the domain index. This can be done using algorithms such as conditional maximum entropy models (Berger et al., 1996) (or multinomial logistic regression) , which benefit from strong theoretical guarantees (Mohri et al., 2018, chapter 13), or other rich models based on neural networks.

Other variants of MULTIBOOST. In Appendix D, we discuss other variants of our algorithm, including their convergence guarantees.

MULTIBOOST$(S_1, \ldots, S_p)$

$\alpha_0 \leftarrow 0$
for $t \leftarrow 1$ **to** T **do**
  $f_{t-1} \leftarrow \sum_{l=1}^{p} \sum_{j=1}^{N_l} \alpha_{t-1,l,j} \mathsf{Q}(l|\cdot)\, h_{l,j}$
  $\tilde{\Phi}_k \leftarrow \frac{1}{m_k} \sum_{i=1}^{m_k} \Phi(-y_{k,i} f_{t-1}(x_{k,i}))$
  $\mathcal{K}_t \leftarrow \left\{k \colon k \in \operatorname{argmax}_{k\in[p]} \tilde{\Phi}_k\right\}$
  for $k \in \mathcal{K}_t$ **do**
    $Z_{t,k} \leftarrow \sum_{i=1}^{m_k} \Phi'\big(-y_{k,i} f_{t-1}(x_{k,i})\big)$
    for $i \leftarrow 1$ **to** m_k **do**
      $\mathsf{D}_t(k,i) \leftarrow \frac{\Phi'(-y_{k,i} f_{t-1}(x_{k,i}))}{Z_{t,k}}$
  $(k, r) \leftarrow \operatorname*{argmax}_{k\in\mathcal{K}_t, r\in[N_k]} \frac{Z_{t,k}}{m_k}[1 - 2\epsilon_{t,k,r}]$
  $\eta_t \leftarrow \operatorname{argmin}_{\eta>0} F(\alpha_{t-1} + \eta \mathbf{e}_{k,r})$
  $\alpha_t \leftarrow \alpha_{t-1} + \eta_t \mathbf{e}_{k,r}$
$f \leftarrow \sum_{l=1}^{p} \sum_{j=1}^{N_l} \alpha_{T,l,j} \mathsf{Q}(l|\cdot) h_{l,j}$
return f

Figure 1: Pseudocode of the MULTIBOOST algorithm. $\epsilon_{t,k,r} = \frac{1}{2}\left[1 - \mathbb{E}_{i\sim \mathsf{D}_t(k,\cdot)}[y_{k,i}\mathsf{Q}(k|x_{k,i})h_{k,r}(x_{k,i})]\right]$ denotes the weighted error of $\mathsf{Q}(k|\cdot)h_{k,r}$.

4 Theoretical analysis

In this section, we present a theoretical analysis of our algorithm, including margin-based learning bounds for multiple-source Q-ensembles.

For any $k \in [p]$, define the family $\mathcal{G}_k$ as follows: $\mathcal{G}_k = \left\{\mathsf{Q}(k|\cdot)\, h \colon h \in \mathcal{H}_k\right\}$. Then, the family of ensemble functions $\mathcal{F}$ that we consider can be defined as $\mathcal{F} = \operatorname{conv}(\bigcup_{k=1}^{p} \mathcal{G}_k)$. For any $\lambda \in \Delta$, let $\overline{\mathcal{D}}_\lambda$ be the distribution defined by $\mathcal{D}_\lambda = \sum_{k=1}^{p} \lambda_k \widehat{\mathcal{D}}_k$, where $\widehat{\mathcal{D}}_k$ is the empirical distribution associated to an i.i.d. sample S_k drawn from $\mathcal{D}_k$. $\overline{\mathcal{D}}_\lambda$ is distinct from the distribution associated to $\mathcal{D}_\lambda$. We seek to derive margin-based bounds for ensemble functions $f \in \mathcal{F}$ with respect to target mixture distributions $\mathcal{D}_\lambda$, while the empirical data is from multiple source distributions $\mathcal{D}_k$, or from $\overline{\mathcal{D}}_\lambda$. Thus, these guarantees differ from standard learning bounds where the source and target distribution coincide.

For any distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$ and $\rho > 0$, let $\mathcal{L}_\rho(\mathcal{D}, f)$ denote the expected ρ-margin loss of f with respect to $\mathcal{D}$: $\mathcal{L}_\rho(\mathcal{D}, f) = \mathbb{E}_{(x,y)\sim\mathcal{D}_k}[\mathbb{I}(yf(x) \leq \rho)]$. We denote by $\mathfrak{R}_m(\mathcal{F})$ the Rademacher complexity of $\mathcal{F}$ for samples $S = (x_1, \ldots, x_m)$ of size m, defined by $\mathfrak{R}_m(\mathcal{F}) = \mathbb{E}_{S\sim\mathcal{D}^m}[\sup_{f\in\mathcal{F}} \sum_{i=1}^{m} \sigma_i f(x_i)]$, with σ_is independent random variables taking values in $\{-1, +1\}$. The following gives a margin-bound for our multiple-source learning with Q-ensembles.

Theorem 1. *Fix $\rho > 0$. Then, for any $\delta > 0$, with probability at least $1 - \delta$ over the draw of a sample $S = (S_1, \ldots, S_p) \sim \mathcal{D}_1^{m_1} \otimes \cdots \otimes \mathcal{D}_p^{m_p}$, the following inequality holds for all ensemble functions $f = \sum_{t=1}^{T} \alpha_t \mathsf{Q}(k_t|\cdot) h_t \in \mathcal{F}$ and all $\lambda \in \Delta$:*

$$\mathcal{L}(\mathcal{D}_\lambda, f) \leq \mathcal{L}_\rho(\overline{\mathcal{D}}_\lambda, f) + \sum_{k=1}^{p} \lambda_k \left[\frac{2}{\rho} \max_{l\in[p]} \mathfrak{R}_{m_k}(\mathcal{H}_l) + \sqrt{\frac{\log \frac{p}{\delta}}{2m_k}}\right]. \tag{5}$$

Note that the complexity term depends only on the Rademacher complexities of the families of based predictors $\mathcal{H}_l$ and does not involve the domain classification function Q. Theorem 1 improves upon previous known bounds of Mohri et al. (2019). In particular, it provides a dimension-independent margin guarantee for the zero-one loss, while the bound of Mohri et al. (2019), when applied to the zero-one loss, is dimension-dependent (Mohri et al., 2019, Lemma 3). Additionally, our bound holds for Q-ensembles. Our margin bound can be straightforwardly generalized to hold uniformly for all $\rho \in (0, 1)$, at the cost of an additional mild term depending on $\log \log_2(2/\rho)$ (see Theorem 4 in Appendix E). For the ensemble functions returned by our algorithm, the bound applies to $x \mapsto \frac{f(x)}{\|\alpha\|_1}$. These learning guarantees suggest choosing $\alpha \geq 0$ and ρ to minimize the following:

$$\max_{\lambda\in\Lambda} \sum_{k=1}^{p} \lambda_k \left\{\mathop{\mathbb{E}}_{(x,y)\sim S_k}\left[\mathbb{I}\left(y \sum_{t=1}^{T} \frac{\alpha_t}{\|\alpha\|_1} \mathsf{Q}(k_t|\cdot) h_t(x) \leq \rho\right)\right] + \frac{r_k}{\rho}\right\},$$

where $r_k = 2\max_{l\in[p]} \mathfrak{R}_{m_k}(\mathcal{H}_l)$. Choosing $\frac{1}{\rho} = \|\alpha\|_1$ and upper bounding the binary loss using a convex function Φ, the problem can then be equivalently cast as the following unconstrained minimization, for example in the case of the exponential function:

$$\min_{\alpha\geq 0}\max_{\lambda\in\Lambda}\sum_{k=1}^{p}\lambda_k\left\{\mathop{\mathbb{E}}_{(x,y)\sim S_k}\left[\exp\left(-y\sum_{t=1}^{T}\alpha_t \mathsf{Q}(k_t|\cdot)h_t(x)\right)\right] + r_k\|\alpha\|_1\right\}.$$

Thus, the learning guarantees justify a posteriori an ℓ_1-regularized version of our algorithm. It is straightforward to derive that extension of our algorithm and its pseudocode. This is similar to the extension to the ℓ_1-ADABOOST (Rätsch et al., 2001) of the original unregularized ADABOOST (Freund and Schapire, 1997). Let us add that finer learning guarantees such as those for DEEP BOOSTING (Cortes et al., 2014) ones can be derived similarly here, which would lead to a weighted ℓ_1-regularization, where the weights are the Rademacher complexities of the families $\mathcal{H}_k$. For the sake of simplicity, we do not detail that extension. Let us point out, however, that such an extension would also lead to an algorithm generalizing ADANET (Cortes et al., 2017) to multiple sources.

In some applications, prior knowledge can be used to constrain Λ to a strict subset of the simplex Δ. Finer margin-based learning guarantees can be derived to cover that scenario. Let $\overline{\Lambda}$ be the set of vertices of a subsimplicial cover of Λ, that is a decomposition of a cover of Λ into subsimplices and let $\overline{\Lambda}_\epsilon$ be the set of λs ϵ-close to $\overline{\Lambda}$ in ℓ_1-distance. Then, the following guarantees hold.

Theorem 2. *Fix $\rho > 0$ and $\epsilon > 0$. Then, for any $\delta > 0$, with probability at least $1-\delta$ over the draw of a sample $S = (S_1, \ldots, S_p) \sim \mathcal{D}_1^{m_1} \otimes \cdots \otimes \mathcal{D}_p^{m_p}$, each of the following inequalities holds:*

1. for all ensemble functions $f = \sum_{t=1}^{T}\alpha_t \mathsf{Q}(k_t|\cdot)h_t \in \mathcal{F}$ and all $\lambda \in \overline{\Lambda}_\epsilon$:

$$\mathcal{L}(\mathcal{D}_\lambda, h) \leq \mathcal{L}_\rho(\overline{\mathcal{D}}_\lambda, h) + \frac{2}{\rho}\max_{r\in[p]}\mathfrak{R}_{\mathbf{m}}(\mathcal{H}_r, \lambda) + \frac{3\epsilon}{2} + \sqrt{\sum_{k=1}^{p}\frac{\lambda_k^2}{2m_k}\log\frac{|\overline{\Lambda}|}{\delta}}. \tag{6}$$

2. for all ensemble functions $f = \sum_{t=1}^{T}\alpha_t \mathsf{Q}(k_t|\cdot)h_t \in \mathcal{F}$ and all $\lambda = \sum_{k=1}^{p}\mu_k\beta_k \in \mathrm{conv}(\overline{\Lambda})$:

$$\mathcal{L}(\mathcal{D}_\lambda, h) \leq \mathcal{L}_\rho(\overline{\mathcal{D}}_\lambda, h) + \frac{2}{\rho}\sum_{l=1}^{p}\mu_k\max_{r\in[p]}\mathfrak{R}_{\mathbf{m}}(\mathcal{H}_r, \beta_l) + \sum_{k=1}^{p}\mu_k\sqrt{\sum_{l=1}^{p}\frac{\beta_{l,k}^2}{2m_k}\log\frac{|\overline{\Lambda}|}{\delta}}. \tag{7}$$

The proof of the theorem and further discussion are presented in Appendix F. Here, $\mathfrak{R}_{\mathbf{m}}(\mathcal{H}_r, \lambda)$ is the λ-weighted Rademacher complexity of $\mathcal{H}_r$ (see Appendix F). For Λ a strict subset of the simplex, these learning bounds suggest an algorithm with a regularization term of the form $\sum_{k=1}^{p}\lambda_k^2/m_k$.

5 Federated MULTIBOOST algorithm

In this section, we present an extension of our algorithm to the *federated learning* scenario, called FEDMULTIBOOST. The pseudocode is given in Figure 2. Federated learning is a distributed learning scenario where a global model is trained at the server level, based on data remaining at clients (Konečný et al., 2016b,a; McMahan et al., 2017; Yang et al., 2019). Clients are typically mobile phones, network sensors, or other IoT devices. While the data remains at the clients, the trained model significantly benefits from user data, as demonstrated, for example, in next word prediction (Hard et al., 2018; Yang et al., 2018) and healthcare applications (Brisimi et al., 2018). We refer the reader to (Kairouz et al., 2021) for a more detailed survey of federated learning.

A widely used algorithm in this scenario is *federated averaging*, where the server trains the global model via stochastic updates from the clients (McMahan et al., 2017). It has been argued by Mohri et al. (2019), however, that minimizing the expected loss with respect to a particular training distribution, as done by standard federated learning algorithms, may be risk-prone and benefit only a specific subset of clients. This compromises fairness, which is one of the critical questions in federated learning, where clients are often heterogeneous (Bickel et al., 1975; Hardt et al., 2016; Abay et al., 2020; Li et al., 2020). This issue can be further accentuated when the set of participating clients during inference may significantly differ from those used in training. Such cases may occur, for example, when clients loose access to network connection during training, which results in distinct training and test distributions.

To overcome these problems, we adopt the *agnostic federated learning* approach suggested by Mohri et al. (2019) and Ro et al. (2021). Following the client-partition approach of Ro et al. (2021), let each client belong to one of p domains. The domains can be, for example, based on the type of device or their geographical location. In this setting, the global model is optimized for all convex combinations of domain distributions as in the objective of MultiBoost (Equation 4), by taking the maximum over the mixtures weights λ. This ensures that the solution obtained is risk-averse and reduces the worst-case mismatch between the inference and training distributions. We propose FedMultiBoost, a communication-efficient modification of MultiBoost, that can overcome the communication bottleneck in federated learning (Konečný et al., 2016b).

Before presenting our algorithm, we first describe the learning scenario and the notation we adopt. Let n be the number of clients, which in the cross-device setting can be in the order of millions. Let m_c denote the number of samples in client c and $(x_{c,i}, y_{c,i})$ denote the ith sample of client c. The server has access to a set of base-predictor classes $\mathcal{H}_k$ for $k \in [p]$ and a domain classifier Q that assigns for each sample x, the likelihood of it belonging to each of the p domains. The base classifiers can be obtained by training on public data or training on client data with federated learning. The domain classifier Q can be trained using unlabeled client data.

FedMultiBoost$(S_1, S_2, \ldots, S_n)$

$\alpha_0 \leftarrow 0$
for $t \leftarrow 1$ **to** T **do**
  $f_{t-1} \leftarrow \sum_{l=1}^{p} \sum_{j=1}^{N_l} \alpha_{t-1,l,j} \mathsf{Q}(l|\cdot)\, h_{l,j}$
  $C_{t,k} \leftarrow$ SelectClient(k, r)
  for $k \leftarrow 1$ **to** p **do**
    for $c \in C_{t,k}$ **do**
      $\tilde{\Phi}_{k,c} \leftarrow \frac{1}{m_c} \sum_{i=1}^{m_c} \Phi(-y_{c,i} f_{t-1}(x_{c,i}))$
      $Z_{t,c} \leftarrow \sum_{i=1}^{m_c} \Phi'\big(-y_{c,i} f_{t-1}(x_{c,i})\big)$
  $\mathcal{K}_t \leftarrow \Big\{k \in [p] \colon k \in \operatorname{argmax}_{k \in [p]} \sum_{c \in C_{t,k}} \tilde{\Phi}_{k,c}\Big\}$
  $Z_{t,k} \leftarrow \sum_{c \in C_{t,k}} Z_{t,c}, \forall k \in [p]$
  $\mathcal{H}_{t,k} \leftarrow$ SelectBaseClassifier(k, s)
  for $k \in \mathcal{K}_t$ **do**
    for $c \in C_{t,k}$ **do**
      for $i \leftarrow 1$ **to** m_c **do**
        $\mathsf{D}_t(c, i) \leftarrow \frac{\Phi'(-y_{c,i} f_{t-1}(x_{c,i}))}{Z_{t,k}}$
      $\beta_{t,c,r} \leftarrow [1 - 2\epsilon_{t,c,r}]$
  $(k, r) \leftarrow \operatorname*{argmax}_{k \in \mathcal{K}_t, r \in \mathcal{H}_{t,k}} \frac{Z_{t,k}}{\sum_{c \in C_{t,k}} m_c} \sum_{c \in C_{t,k}} \beta_{t,c,r}$
  $\eta_t \leftarrow \eta_0 / \sqrt{t}$
  $\alpha_t \leftarrow \alpha_{t-1} + \eta_t \mathbf{e}_{k,r}$
$f \leftarrow \sum_{l=1}^{p} \sum_{j=1}^{N_l} \alpha_{T,l,j} \mathsf{Q}(l|\cdot) h_{l,j}$
return f

Figure 2: Pseudocode of the FedMultiBoost algorithm. $\epsilon_{t,c,r}$ denotes the weighted error of $\mathsf{Q}(k|\cdot)h_{k,r}$, $\epsilon_{t,c,r} = \frac{1}{2}\big[1 - \mathbb{E}_{i \sim \mathsf{D}_t(c,\cdot)}[y_{c,i} \mathsf{Q}(k|x_{c,i}) h_{k,r}(x_{c,i})]\big]$, where client c belongs to domain k. cobalt steps are carried out by the server, red on the clients.

Additional discussion of the algorithm as well as experimental results can be found in Appendix G. At each round, the algorithm randomly selects r clients and s base classifiers from each domain. Next, it chooses the best classifier out of the previously selected subset, based on the data from the subsampled clients. Since FedMultiBoost operates only on a small set of base classifiers at each round, the algorithm is communication-efficient. At round t, the server needs to send Q, f_{t-1}, and s base classifiers to the selected clients. Hence, the server-to-client communication cost at round t is $\mathcal{O}((t+s)K + K')$, where K is the number of parameters in a single base classifier and K' is the number of parameters required to specify Q. Furthermore, the clients communicate only the aggregate statistics $\tilde{\Phi}_c, Z_{t,c}, \beta_{t,c,r}$ to the server and hence the client-to-server communication is $\mathcal{O}(p \cdot s)$ real numbers, which is independent of the number of parameters of the base classifiers.

6 Experiments

In this section, we present experimental results for the MultiBoost algorithm on several multiple-source datasets. Our study is restricted to learning an ensemble of decision stumps $\mathcal{H}^{stumps}$ using the exponential surrogate loss $\Phi(u) = e^{-u}$. To estimate the probabilities $\mathsf{Q}(k|\cdot)$ for $k \in [p]$, we assigned the label k to each sample from domain k and used multinomial logistic regression. This step can be facilitated with unlabeled examples, as only their domain membership information is required.

We compare MultiBoost with a set of boosting-style algorithms that operate on the same hypotheses class $\mathcal{H}^{stumps}$. Those include AdaBoost-k for $k \in [p]$ and AdaBoost-all. The former is a

standard ADABOOST algorithm trained only on a single source S_k and the latter is ADABOOST trained on the union of all sources $\cup_{k=1}^{p} S_k$. It is natural to compare MULTIBOOST against ADABOOST-all, since both of them have access to all sources during training. The difference is that while ADABOOST-all treats $\cup_{k=1}^{p} S_k$ as a single dataset, MULTIBOOST trains base learners separately for each source and weights examples by domain probabilities $\mathsf{Q}(k|\cdot)$. We used $T = 100$ boosting steps for all benchmarks. T up to 1,000 were explored, but did not change any results significantly.

We report classification errors on the test data for various mixtures of the source distributions $\lambda_1, \ldots, \lambda_p$, including: errors for λ on the simplex Δ edges (errors on each domain separately); errors on the uniform mixture $\forall k : \lambda_k = \frac{1}{p}$; agnostic error, which is the maximum error across all sources. The errors and their standard deviations are reported based on 10-fold cross validation. Each source S_k is independently split into 10 folds $S_k^1, \ldots, S_k^{10}$. For the i-th cross-validation step, the test set is $\{S_1^i, \ldots, S_p^i\}$, while the rest is used for training.

Datasets and preprocessing steps used are described below with additional dataset details provided in Appendix H. Note, that all datasets are public and do not contain any personal identifiers or offensive information. The experiments were performed on Linux and Mac workstations with Quad-Core Intel Core i7 2.9 GHz and Intel Xeon 2.20 GHz respectively. The time taken to perform 10-fold cross-validation varies per dataset: from roughly 1 hour on CPU for all benchmarks on tabular data to 8 hours on CPU for all benchmarks on digits data. The run-times of a single MULTIBOOST iteration is almost identical to that of ADABOOST-all when a closed form solution is used for step size η_t (line 12 in Figure 1) for the exponential surrogate loss function. Alternatively, for some experiments we used line search with 100 steps. If training time is critical, one can use $\eta_t = C/\sqrt{t}$ instead, at the cost of slightly worse convergence guarantee. Convergence plots are illustrated in Appendix I.

Sentiment analysis. The *Sentiment* dataset (Blitzer et al., 2007) consists of text reviews and rating labels for products sold on Amazon in various categories. We selected $p = 3$ sources from this dataset, where $k = 1$ corresponds to `books`, $k = 2$ means `dvd` and $k = 3$ is `electronics`. We converted the labels to binary values, assigning $y = +1$ for a positive review and $y = -1$ for a negative one. The benchmarks are trained using bigram features generated from the raw text reviews. Since the *Sentiment* dataset contains only 2,000 instances for each domain, we used random train/test splits with 10 different seeds instead of the 10-fold cross-validation. To illustrate the importance of the conditional domain probabilities $\mathsf{Q}(K|x)$, for each domain, we illustrate its values along the projections of x onto the first principal component of the joint *Sentiment* dataset in Figure 3. The figure shows that our domain classifiers are able to provide coherent separation between the three domains and narrow the applicability of the base classifiers.

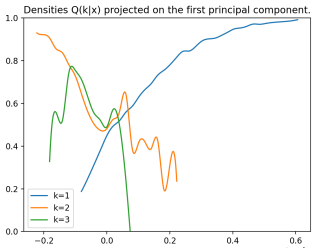

Figure 3: Mean values of $\mathsf{Q}(k|\cdot_k)$ for the three domains of the Sentiment data projected on the first principal component of the joint data.

Digits recognition. For this problem, we aggregated 32x32 pixel handwritten digits images from $p = 3$ sources: *MNIST* ($k = 1$), *SVHN* ($k = 2$), *MNIST-M* ($k = 3$). We compared algorithms on two binary classification problems: digits 4 vs. 9 and digits 1 vs. 7.

Object recognition. We divided images of clothes items from *Fashion-MNIST* (Xiao et al., 2017) dataset into two classes: tops and bottoms from $p = 3$ sources. $k = 1$ consists of `t-shirts, pullovers, trousers`; $k = 2$ consists of `dresses, coats, sandals`; $k = 3$ consists of `shirts, bags, sneakers, ankle-boots`.

Tabular data. In addition to text and images, we tested MULTIBOOST on tabular data. We used the *Adult* dataset (Kohavi, 1996), which consists of numerical and categorical features describing an individual's socioeconomic characteristics, given the task to predict if an person's income exceeds USD $50,000$. We divided this dataset into $p = 3$ sources based on individual's educational background. Source $k = 1$ consisted of individuals with a university degree (BSc, MS or PhD), source $k = 1$ those with only a High School diploma, and source $k = 2$, none of the above.

Discussion As can be seen from Table 1, MULTIBOOST provides agnostic and uniform errors that are significantly better than the baselines on all datasets. While ADABOOST-k predictors on digits recognition and object recognition tasks each show the lowest error on their specific domains $\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3$, they fail to generalize to other target distribution in $\mathrm{conv}\{\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3\}$. Moreover, as suggested by the discussion in Section 2, the standard convex combination of domain-specific predictors

Table 1: Test errors for multiple benchmarks.

Algorithm	Error-1	Error-2	Error-3	Error-Uniform	Error-Agnostic
			Sentiment Analysis		
ADABOOST-1	**0.326** $\pm$ **0.019**	0.300 $\pm$ 0.008	0.360 $\pm$ 0.017	0.329 $\pm$ 0.017	0.360 $\pm$ 0.019
ADABOOST-2	0.354 $\pm$ 0.020	**0.266** $\pm$ **0.009**	0.336 $\pm$ 0.023	0.318 $\pm$ 0.019	0.357 $\pm$ 0.019
ADABOOST-3	0.402 $\pm$ 0.015	0.334 $\pm$ 0.008	**0.258** $\pm$ **0.015**	0.331 $\pm$ 0.016	0.402 $\pm$ 0.018
ADABOOST-all	0.354 $\pm$ 0.020	0.325 $\pm$ 0.011	0.313 $\pm$ 0.022	0.324 $\pm$ 0.021	0.354 $\pm$ 0.016
MULTIBOOST	0.332 $\pm$ 0.027	0.288 $\pm$ 0.018	0.284 $\pm$ 0.027	**0.301** $\pm$ **0.027**	**0.332** $\pm$ **0.024**
			Digits Recognition (4 vs. 9)		
ADABOOST-1	**0.044** $\pm$ **0.007**	0.615 $\pm$ 0.012	0.476 $\pm$ 0.022	0.379 $\pm$ 0.008	0.615 $\pm$ 0.012
ADABOOST-2	0.455 $\pm$ 0.014	0.299 $\pm$ 0.011	0.504 $\pm$ 0.015	0.420 $\pm$ 0.011	0.504 $\pm$ 0.015
ADABOOST-3	0.549 $\pm$ 0.034	0.488 $\pm$ 0.015	0.300 $\pm$ 0.013	0.446 $\pm$ 0.013	0.549 $\pm$ 0.034
ADABOOST-all	0.060 $\pm$ 0.009	0.374 $\pm$ 0.015	0.353 $\pm$ 0.012	0.262 $\pm$ 0.009	0.374 $\pm$ 0.015
MULTIBOOST	0.096 $\pm$ 0.008	**0.283** $\pm$ **0.028**	**0.246** $\pm$ **0.014**	**0.209** $\pm$ **0.013**	**0.284** $\pm$ **0.027**
			Digits Recognition (1 vs. 7)		
ADABOOST-1	**0.005** $\pm$ **0.002**	0.613 $\pm$ 0.007	0.519 $\pm$ 0.012	0.379 $\pm$ 0.004	0.613 $\pm$ 0.007
ADABOOST-2	0.431 $\pm$ 0.022	**0.252** $\pm$ **0.009**	0.479 $\pm$ 0.012	0.387 $\pm$ 0.010	0.479 $\pm$ 0.012
ADABOOST-3	0.680 $\pm$ 0.031	0.490 $\pm$ 0.014	**0.244** $\pm$ **0.012**	0.474 $\pm$ 0.013	0.680 $\pm$ 0.031
ADABOOST-all	0.014 $\pm$ 0.003	0.286 $\pm$ 0.010	0.306 $\pm$ 0.012	0.202 $\pm$ 0.005	0.306 $\pm$ 0.011
MULTIBOOST	0.026 $\pm$ 0.004	0.261 $\pm$ 0.013	0.257 $\pm$ 0.015	**0.181** $\pm$ **0.005**	**0.261** $\pm$ **0.011**
			Objects Recognition (Fashion-MNIST)		
ADABOOST-1	**0.015** $\pm$ **0.003**	0.251 $\pm$ 0.026	0.602 $\pm$ 0.028	0.288 $\pm$ 0.017	0.602 $\pm$ 0.028
ADABOOST-2	0.435 $\pm$ 0.007	**0.015** $\pm$ **0.002**	0.169 $\pm$ 0.012	0.173 $\pm$ 0.003	0.435 $\pm$ 0.007
ADABOOST-3	0.311 $\pm$ 0.018	0.097 $\pm$ 0.005	**0.014** $\pm$ **0.002**	0.140 $\pm$ 0.006	0.311 $\pm$ 0.018
ADABOOST-all	0.036 $\pm$ 0.004	0.020 $\pm$ 0.002	0.025 $\pm$ 0.003	0.027 $\pm$ 0.002	0.036 $\pm$ 0.004
MULTIBOOST	0.028 $\pm$ 0.003	0.015 $\pm$ 0.003	0.022 $\pm$ 0.002	**0.021** $\pm$ **0.001**	**0.028** $\pm$ **0.003**
			Tabular Data (Adult Data)		
ADABOOST-1	0.201 $\pm$ 0.012	0.249 $\pm$ 0.021	0.215 $\pm$ 0.014	0.222 $\pm$ 0.012	0.249 $\pm$ 0.021
ADABOOST-2	0.298 $\pm$ 0.011	**0.131** $\pm$ **0.009**	0.142 $\pm$ 0.006	0.190 $\pm$ 0.007	0.298 $\pm$ 0.011
ADABOOST-3	0.225 $\pm$ 0.015	0.138 $\pm$ 0.010	0.133 $\pm$ 0.011	0.165 $\pm$ 0.008	0.225 $\pm$ 0.015
ADABOOST-all	0.221 $\pm$ 0.013	0.134 $\pm$ 0.007	0.131 $\pm$ 0.010	0.155 $\pm$ 0.004	0.221 $\pm$ 0.013
MULTIBOOST	**0.190** $\pm$ **0.014**	0.132 $\pm$ 0.008	**0.130** $\pm$ **0.009**	**0.150** $\pm$ **0.005**	**0.190** $\pm$ **0.014**

(ADABOOST-all) also performs poorly on several target distributions in $\mathrm{conv}\{\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3\}$, such as the uniform and agnostic targets. As the experimental results confirm, MULTIBOOST, by leveraging domain predictors $\mathsf{Q}(k|\cdot)$ and the agnostic loss, is able to produce an ensemble of domain-specific predictors that generalizes to different targets. Appendix I provides additional illustrations of $\mathsf{Q}(k|\cdot)$ by projecting each domain on the first principal component of the joint data.

Moreover, for tabular data and the digits (4 vs. 9) classification problem, MULTIBOOST benefits from positive knowledge transfer and obtains an error on individual domains that is even smaller than that of the domain-specific predictors. From Figure 6 in Appendix I it can be seen that the α-weight for the MULTIBOOST classifier in the 4 vs. 9 problem is heavily centered on $\mathcal{D}_2$, but contributions from $\mathcal{D}_1$ and $\mathcal{D}_3$ adds to the stronger performance. The same figure also illustrates the α-mass distribution for the other classification tasks. The standard convex ensemble ADABOOST-all, however, does not seem to exhibit the positive transfer property. Similar conclusions carry over for other target distributions in $\mathrm{conv}\{\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3\}$, for the sake of simplicity we do not present them here.

7 Conclusion

We presented a new boosting algorithm, MULTIBOOST, for a multiple-source scenario – a common setting in domain adaptation problems – where the target distribution can be any mixture of the source distributions. We showed that our algorithm benefits from strong theoretical guarantees and exhibits favorable empirical performance. We also highlighted the extension of our work to the federated learning scenario, which is a critical distributed learning setting in modern applications.

References

A. Abay, Y. Zhou, N. Baracaldo, S. Rajamoni, E. Chuba, and H. Ludwig. Mitigating bias in federated learning, 2020.

C. J. Becker, C. M. Christoudias, and P. Fua. Non-linear domain adaptation with boosting. In *Proceedings of Advances in Neural Information Processing Systems 26: 27th Annual Conference on NIPS*, pages 485–493, 2013.

S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira. Analysis of representations for domain adaptation. In *Advances in neural information processing systems*, pages 137–144, 2007.

S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan. A theory of learning from different domains. *Machine learning*, 79(1-2):151–175, 2010.

A. L. Berger, S. D. Pietra, and V. J. D. Pietra. A maximum entropy approach to natural language processing. *Comp. Linguistics*, 22(1), 1996.

P. J. Bickel, E. A. Hammel, and J. W. O'Connell. Sex bias in graduate admissions: Data from Berkeley. *Science*, 187(4175):398–404, 1975. ISSN 0036-8075.

J. Blitzer, M. Dredze, and F. Pereira. Biographies, Bollywood, Boom-boxes and Blenders: Domain Adaptation for Sentiment Classification. In *Proceedings of ACL 2007*, Prague, Czech Republic, 2007.

J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. Wortman. Learning bounds for domain adaptation. In *Advances in neural information processing systems*, pages 129–136, 2008.

K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konečnỳ, S. Mazzocchi, H. B. McMahan, et al. Towards federated learning at scale: System design. *arXiv preprint arXiv:1902.01046*, 2019.

L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

L. Breiman. Prediction games and arcing algorithms. *Neural Computation*, 11:1493–1517, October 1999.

T. S. Brisimi, R. Chen, T. Mela, A. Olshevsky, I. C. Paschalidis, and W. Shi. Federated learning of predictive models from federated electronic health records. *International journal of medical informatics*, 112:59–67, 2018.

S. Caldas, P. Wu, T. Li, J. Konečnỳ, H. B. McMahan, V. Smith, and A. Talwalkar. Leaf: A benchmark for federated settings. *arXiv preprint arXiv:1812.01097*, 2018.

Y. Cheng, G. Cao, X. Wang, and J. Pan. Weighted multi-source tradaboost. *Chinese Journal of Electronics*, 22(3):505–510, 2013.

C. Cortes and M. Mohri. Domain adaptation and sample bias correction theory and algorithm for regression. *Theoretical Computer Science*, 519:103–126, 2014.

C. Cortes, M. Mohri, and U. Syed. Deep boosting. In *Proceedings of ICML*, pages 1179–1187, 2014.

C. Cortes, M. Mohri, and A. Muñoz Medina. Adaptation algorithm and theory based on generalized discrepancy. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 169–178, 2015.

C. Cortes, X. Gonzalvo, V. Kuznetsov, M. Mohri, and S. Yang. Adanet: Adaptive structural learning of artificial neural networks. In *Proceedings of ICML*, pages 874–883, 2017.

K. Crammer, M. J. Kearns, and J. Wortman. Learning from multiple sources. *Journal of Machine Learning Research*, 9(Aug):1757–1774, 2008.

W. Dai, Q. Yang, G.-R. Xue, and Y. Yu. Boosting for transfer learning. In *Proceedings of ICML*, pages 874–883, 2007.

Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer System Sciences*, 55(1):119–139, 1997.

Y. Freund, Y. Mansour, and R. E. Schapire. Generalization bounds for averaged classifiers. *The Annals of Statistics*, 32:1698–1722, 2004.

Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR, 2015.

R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *2012 IEEE conference on computer vision and pattern recognition*, pages 2066–2073. IEEE, 2012.

A. Habrard, J.-P. Peyrache, and M. Sebban. Boosting for unsupervised domain adaptation. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 433–448. Springer, 2013.

J. Hamer, M. Mohri, and A. T. Suresh. Fedboost: A communication-efficient algorithm for federated learning. In *International Conference on Machine Learning*, pages 3973–3983. PMLR, 2020.

J. Hampshire and A. Waibel. The meta-pi network: building distributed knowledge representations for robust multisource pattern recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(7):751–769, 1992. doi: 10.1109/34.142911.

A. Hard, K. Rao, R. Mathews, F. Beaufays, S. Augenstein, H. Eichner, C. Kiddon, and D. Ramage. Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604*, 2018.

M. Hardt, E. Price, E. Price, and N. Srebro. Equality of opportunity in supervised learning. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper/2016/file/9d2682367c3935defcb1f9e247a97c0d-Paper.pdf.

J. Hoffman, B. Kulis, T. Darrell, and K. Saenko. Discovering latent domains for multisource domain adaptation. In *ECCV*, volume 7573, pages 702–715, 2012.

J. Hoffman, E. Rodner, J. Donahue, T. Darrell, and K. Saenko. Efficient learning of domain-invariant image representations. *arXiv preprint arXiv:1301.3224*, 2013.

J. Hoffman, M. Mohri, and N. Zhang. Algorithms and theory for multiple-source adaptation. In *Proceedings of NeurIPS*, pages 8256–8266, 2018.

P. Huang, G. Wang, and S. Qin. A novel learning approach to multiple tasks based on boosting methodology. *Pattern recognition letters*, 31(12):1693–1700, 2010.

P. Huang, G. Wang, and S. Qin. Boosting for transfer learning from multiple data sources. *Pattern Recognition Letters*, 33(5):568–579, 2012.

R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991.

P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, R. G. L. D'Oliveira, H. Eichner, S. E. Rouayheb, D. Evans, J. Gardner, Z. Garrett, A. Gascón, B. Ghazi, P. B. Gibbons, M. Gruteser, Z. Harchaoui, C. He, L. He, Z. Huo, B. Hutchinson, J. Hsu, M. Jaggi, T. Javidi, G. Joshi, M. Khodak, J. Konečný, A. Korolova, F. Koushanfar, S. Koyejo, T. Lepoint, Y. Liu, P. Mittal, M. Mohri, R. Nock, A. Özgür, R. Pagh, M. Raykova, H. Qi, D. Ramage, R. Raskar, D. Song, W. Song, S. U. Stich, Z. Sun, A. T. Suresh, F. Tramèr, P. Vepakomma, J. Wang, L. Xiong, Z. Xu, Q. Yang, F. X. Yu, H. Yu, and S. Zhao. Advances and open problems in federated learning, 2021.

R. Kohavi. Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid. In *Kdd*, volume 96, pages 202–207, 1996.

J. Konečnỳ, H. B. McMahan, D. Ramage, and P. Richtárik. Federated optimization: Distributed machine learning for on-device intelligence. *arXiv preprint arXiv:1610.02527*, 2016a.

J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016b.

N. Konstantinov and C. Lampert. Robust learning from untrusted sources. In *International Conference on Machine Learning*, pages 3488–3498, 2019.

P. Lahoti, A. Beutel, J. Chen, K. Lee, F. Prost, N. Thain, X. Wang, and E. H. Chi. Fairness without demographics through adversarially reweighted learning. *arXiv e-prints*, pages arXiv–2006, 2020.

M. Ledoux and M. Talagrand. *Probability in Banach Spaces: Isoperimetry and Processes*. Springer, New York, 1991.

D. Levy, Y. Carmon, J. C. Duchi, and A. Sidford. Large-scale methods for distributionally robust optimization. *Advances in Neural Information Processing Systems*, 33, 2020.

T. Li, M. Sanjabi, A. Beirami, and V. Smith. Fair resource allocation in federated learning. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=ByexElSYDr.

H. Liao. Speaker adaptation of context dependent deep neural networks. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 7947–7951. IEEE, 2013.

M. Long, Y. Cao, J. Wang, and M. Jordan. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pages 97–105. PMLR, 2015.

Z.-Q. Luo and P. Tseng. On the convergence of the coordinate descent method for convex differentiable minimization. *Journal of Optimization Theory and Applications*, 72(1):7–35, 1992.

D. J. C. MacKay. *Bayesian methods for adaptive models*. PhD thesis, California Institute of Technology, 1991.

Y. Mansour, M. Mohri, and A. Rostamizadeh. Domain adaptation with multiple sources. *Advances in neural information processing systems*, 21:1041–1048, 2008.

Y. Mansour, M. Mohri, and A. Rostamizadeh. Multiple source adaptation and the Rényi divergence. In *UAI*, pages 367–374, 2009a.

Y. Mansour, M. Mohri, and A. Rostamizadeh. Domain adaptation: Learning bounds and algorithms. In *COLT*, 2009b.

B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of AISTATS*, pages 1273–1282, 2017.

M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations of Machine Learning*. Adaptive computation and machine learning. MIT Press, 2018. Second edition.

M. Mohri, G. Sivek, and A. T. Suresh. Agnostic federated learning. In *International Conference on Machine Learning*, pages 4615–4625. PMLR, 2019.

S. Motiian, Q. Jones, S. Iranmanesh, and G. Doretto. Few-shot adversarial domain adaptation. In *Advances in Neural Information Processing Systems*, pages 6670–6680, 2017a.

S. Motiian, M. Piccirilli, D. A. Adjeroh, and G. Doretto. Unified deep supervised domain adaptation and generalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5715–5725, 2017b.

K. Muandet, D. Balduzzi, and B. Schölkopf. Domain generalization via invariant feature representation. In *International Conference on Machine Learning*, pages 10–18. PMLR, 2013.

H. Namkoong and J. C. Duchi. Stochastic gradient methods for distributionally robust optimization with f-divergences. In *Advances in Neural Information Processing Systems*, pages 2208–2216, 2016.

S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

Z. Pei, Z. Cao, M. Long, and J. Wang. Multi-adversarial domain adaptation. In *AAAI*, pages 3934–3941, 2018.

G. Rätsch, S. Mika, and M. K. Warmuth. On the convergence of leveraging. In *NIPS*, pages 487–494, 2001.

J. Ro, M. Chen, R. Mathews, M. Mohri, and A. T. Suresh. Communication-efficient agnostic federated averaging. *arXiv preprint arXiv:2104.02748*, 2021.

K. Saito, D. Kim, S. Sclaroff, T. Darrell, and K. Saenko. Semi-supervised domain adaptation via minimax entropy. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8050–8058, 2019.

R. E. Schapire, Y. Freund, P. Bartlett, and W. S. Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. In *ICML*, pages 322–330, 1997.

Z. Shen, H. Hassani, S. Kale, and A. Karbasi. Federated functional gradient boosting. *arXiv preprint arXiv:2103.06972*, 2021.

P. Smyth and D. Wolpert. Linearly combining density estimators via stacking. *Machine Learning*, 36: 59–83, July 1999.

A. Taherkhani, G. Cosma, and T. M. McGinnity. Adaboost-cnn: An adaptive boosting algorithm for convolutional neural networks to classify multi-class imbalanced datasets using transfer learning. *Neurocomputing*, 404:351–366, 2020.

A. Torralba and A. A. Efros. Unbiased look at dataset bias. In *CVPR 2011*, pages 1521–1528. IEEE, 2011.

P. Tseng. Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal Optimization Theory and Applications*, pages 475–494, 2001.

E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko. Simultaneous deep transfer across domains and tasks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4068–4076, 2015.

B. Wang, J. Mendez, M. Cai, and E. Eaton. Transfer learning via minimizing the performance gap between domains. In *Advances in Neural Information Processing Systems*, pages 10645–10655, 2019a.

T. Wang, X. Zhang, L. Yuan, and J. Feng. Few-shot adaptive faster r-cnn. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7173–7182, 2019b.

H. Xiao, K. Rasul, and R. Vollgraf. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, abs/1708.07747, 2017. URL http://arxiv.org/abs/1708.07747.

R. Xu, Z. Chen, W. Zuo, J. Yan, and L. Lin. Deep cocktail network: Multi-source unsupervised domain adaptation with category shift. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3964–3973, 2018.

Z. Xu and S. Sun. Multi-view transfer learning with adaboost. In *2011 IEEE 23rd International Conference on Tools with Artificial Intelligence*, pages 399–402. IEEE, 2011.

Z. Xu and S. Sun. Multi-source transfer learning with multi-view adaboost. In *International conference on neural information processing*, pages 332–339. Springer, 2012.

Z. Xu, W. Li, L. Niu, and D. Xu. Exploiting low-rank structure from latent domains for domain generalization. In *European Conference on Computer Vision*, pages 628–643. Springer, 2014.

J. Yang, R. Yan, and A. G. Hauptmann. Cross-domain video concept detection using adaptive svms. In *Proceedings of the 15th ACM international conference on Multimedia*, pages 188–197, 2007.

Q. Yang, Y. Liu, T. Chen, and Y. Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–19, 2019.

T. Yang, G. Andrew, H. Eichner, H. Sun, W. Li, N. Kong, D. Ramage, and F. Beaufays. Applied federated learning: Improving google keyboard query suggestions. *arXiv preprint arXiv:1812.02903*, 2018.

Y. Yao and G. Doretto. Boosting for transfer learning with multiple sources. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 1855–1862. IEEE, 2010.

Z. Yuan, D. Bao, Z. Chen, and M. Liu. Integrated transfer learning algorithm using multi-source tradaboost for unbalanced samples classification. In *2017 international conference on computing intelligence and information system (CIIS)*, pages 188–195. IEEE, 2017.

Q. Zhang, H. Li, Y. Zhang, and M. Li. Instance transfer learning with multisource dynamic tradaboost. *The scientific world journal*, 2014, 2014.

H. Zhao, S. Zhang, G. Wu, J. M. Moura, J. P. Costeira, and G. J. Gordon. Adversarial multiple source domain adaptation. In *Advances in neural information processing systems*, pages 8559–8570, 2018.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Sections 5, 7.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Sections 5, 7.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] each Theorem/Lemma lists assumptions
   (b) Did you include complete proofs of all theoretical results? [Yes] See proofs in the Appendix
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Included Pseudocode and URL to datasets in Appendix.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 6
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] 10-fold CV error bars.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Section 6
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] Creators of datasets cited.
   (b) Did you mention the license of the assets? [N/A] Datasets are public, no licence needed.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] URL provided in Appendix
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] Data is public

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] See Section 6, no offensive content or personal ids.

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A] No crowdsourcing or human subjects.

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A] No crowdsourcing or human subjects.

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A] No crowdsourcing or human subjects.